
UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES
PROGRAMA DE EDUCAÇÃO TUTORIAL
GRUPO PET-TELE



Tutorial de Introdução a Banco de Dados
(Versão: A2010M12D16)

Autor: Juliana Geremia

Niterói - RJ
Dezembro / 2010

Sumário

1	Introdução	3
2	O que é um banco de dados	4
2.1	Porque usar banco de dados	4
3	Objetivos de um banco de dados	5
3.1	Acesso rápido aos dados	5
3.2	Redução de redundância e inconsistência de dados	5
3.3	Compartilhamento de dados	5
3.4	Aplicação de restrições de segurança	5
4	Estrutura geral de um sistema de banco de dados	6
4.1	Componentes de processamento de consultas	6
4.1.1	Compilador DML	6
4.1.2	Pré-compilador para comandos DML	6
4.1.3	Interpretador DDL	6
4.1.4	Componentes para o tratamento de consultas	6
4.2	Componentes de autorização e integridade	6
4.2.1	Gerenciamento de autorização e integridade	6
4.2.2	Gerenciamento de transações	6
4.2.3	Administração de arquivos	6
4.2.4	Administração de <i>buffer</i>	7
4.2.5	Arquivo de dados	7
4.2.6	Dicionário de dados	7
4.2.7	Índices	7
4.2.8	Estatísticas de dados	7
5	Software Gerenciador de Banco de Dados	8
5.1	O que é um SGBD	8
5.2	Funcionamento de um SGBD	8
5.2.1	Interação com o gerenciador de arquivos	8
5.2.2	Cumprimento de integridade	8
5.2.3	Cumprimento de segurança	8
5.2.4	Cópias de reserva (<i>backup</i>) e recuperação	9
5.2.5	Controle de concorrência	9
5.3	SGBDs para máquinas pessoais	9
5.4	As interfaces do SGBD	9
5.4.1	Interfaces baseadas em menus para os clientes web ou navegação	9
5.4.2	Interfaces baseadas em formulários	9
5.4.3	Interfaces gráficas para o usuário	9
5.4.4	Interface de linguagem natural	10
5.4.5	Interface para usuários parametrizáveis	10
5.4.6	Interfaces para o administrador do banco de dados	10
5.5	Quando não usar o SGBD	10
5.6	Alguns exemplos de SGBDs	11

6	Modelos de dados	12
6.1	Modelos lógicos baseados em objetos	12
6.1.1	Modelo entidade-relacionamento	12
6.1.2	Modelo orientado a objetos	12
6.2	Modelos lógicos baseados em registros	12
6.2.1	Modelo relacional	12
6.2.2	Modelo hierárquico	12
6.2.3	Modelo de rede	13
6.3	Modelos físicos de dados	13
7	Tipos de Banco de Dados	14
7.1	Banco de Dados Relacional	14
7.1.1	Exemplos de banco de dados, descrição de algumas tabela e seus possíveis atributos	14
7.2	Banco de dados Hierárquico	14
7.3	Banco de dados de rede	15
7.4	Banco de dados Objeto-Relacional	15
8	Linguagens de banco de dados	16
8.1	Linguagem de definições de dados	16
8.2	Linguagem de manipulação de dados	16
9	Usuários do banco de dados	17
9.1	Usuários finais causais	17
9.2	Iniciantes ou usuários finais parametrizáveis	17
9.3	Usuários mais sofisticados	17
9.4	Usuários autônomos	17
10	Administrador do banco de dados	18

1 Introdução

Os bancos de dados se tornaram componentes essenciais no cotidiano da sociedade moderna. No decorrer do dia-a-dia, nos deparamos com atividades que envolvem alguma interação com banco de dados. Nas transações bancárias, nas reservas de passagens aéreas, nas compras de supermercados, no acesso ao catálogo de uma biblioteca informatizada um banco de dados será acessado. O uso de banco de dados facilita o dia-a-dia de quem está envolvido com esse tipo de transação, pois torna-as mais rápidas, eficientes e seguras, e claro, importando em menos trabalho braçal.

Aliados aos bancos de dados, há os *softwares* gerenciadores de banco de dados, um conjunto de programas para acesso aos dados, que tem por objetivo tornar a recuperação e armazenamento dos dados mais eficiente e permitir ao usuário interagir com o banco de dados.

2 O que é um banco de dados

Um banco de dados é uma coleção de dados ou registros relacionados. Esses registros ou dados são fatos que podem ser gravados e que possuem um significado implícito representando aspectos do mundo real. Um banco de dados é projetado, construído e povoado por dados atendendo a uma proposta específica, ou seja, segundo a necessidade do usuário ou de um grupo de usuários. Possui alguns níveis de interação com o mundo real e um público efetivamente interessado em seu conteúdo.

Pode ser de qualquer tamanho e de complexidade variável. Um exemplo de um banco de dados simples e menos complexo seria uma agenda telefônica, na qual provavelmente os registros feitos nesse banco de dados seriam o nome, o endereço, e claro, o telefone de um grupo de pessoas. Mas podemos ter bancos de dados maiores e mais complexos, como por exemplo, os registros de uma biblioteca que tenha por exemplo, um milhão de exemplares de livros, onde esses livros poderiam ter diferentes dados registrados como o título, o autor, o assunto, a edição, o ano de impressão, o número de páginas, o código de acesso ao livro, sua localização na biblioteca e outros dados que a biblioteca achar necessário. Para se ter uma idéia do tamanho que um banco de dados pode ter, ELMASRI e NAVATHE (2005) citam como exemplo o banco de dados mantido pelo *International Revenue Service* (IRS), órgão responsável pelo controle dos formulários de impostos preenchidos pelos contribuintes dos Estados Unidos. Pressupondo que existem cerca de cem milhões de contribuintes e que cada contribuinte preenche a cada ano em média cinco formulários de 400 caracteres cada, é necessário um banco de dados de 100×10^6 (contribuintes) \times 5 (formulários) \times 400 (caracteres (*bytes*) cada formulário) caracteres (*bytes*). Ou seja, um total de 200 *gigabytes*. Um número considerável de dados! A necessidade de se utilizar um banco de dados é evidente!

2.1 Porque usar banco de dados

A resposta é simples: porque facilita a vida das pessoas interessados nos dados registrados em banco de dados pois é compacto, rápido, importa em menos trabalho braçal e tem disponibilidade de acesso as informações corretas e atualizadas a qualquer momento. Basta retomarmos o exemplo anterior dos formulários dos contribuintes estadunidenses, se todos os dados fossem armazenados em papel, seria necessário uma sala enorme para guardar todos os formulários, onde poderiam estar sujeitos a umidade, as traças, ao desaparecimento da tinta no papel, ao roubo, claro que devido ao interesse que há sobre o imposto de renda, eles seriam bem protegidos, mas se pensarmos no armazenamento digital desses dados, com certeza, tudo isso seria evitado, há outros riscos, claro, no armazenamento digital, mas que não se sobrepõem ao do armazenamento em papel. Mas o fato mais interessante do armazenamento digital é como podemos tratar esses dados. Primeiramente, podemos armazená-los diretamente em um banco de dados. Estando os dados guardados em um banco de dados, posteriormente, quando for necessário fazer uma consulta, essa consulta será mais rápida. Tomando como exemplo uma busca por um conjunto de dados, busca essa que pode ser feita em segundos, ou minutos, em um banco de dados digitalizado, dependendo do desempenho da máquina utilizada, enquanto que, se a mesma busca fosse feita manualmente, poderia levar meses até todos os dados de interesse serem computados, o que exigiria um trabalho braçal muito grande e com uma certa probabilidade de haver erros.

O exemplo acima é apenas um dos muitos exemplos que podem ser citados. Podemos descrever de forma mais genérica as vantagens da utilização de bancos de dados na seção “Objetivos de um banco de dados”.

3 Objetivos de um banco de dados

3.1 Acesso rápido aos dados

Um sistema convencional de arquivos não atende de modo eficiente a recuperação de dados. Porém, num sistema de banco de dados, basta escrever o programa necessário para realizar a consulta e obtém-se o resultado desejado em um tempo muito pequeno. Por exemplo, uma empresa deseja saber em qual região a densidade de seus clientes é maior. Se for feita uma busca por CEP nos arquivos da empresa, essa busca terá de ser feita manualmente, ou seja, o funcionário encarregado dessa pesquisa terá de separar manualmente cada cliente por CEP. Isso pode levar um tempo muito grande. Porém, se a busca for feita através de um banco de dados, esse tempo é reduzido consideravelmente, pois podemos ter acesso a todos os clientes de um mesmo CEP de uma única vez. Além disso, a probabilidade de erro diminui consideravelmente.

3.2 Redução de redundância e inconsistência de dados

Num sistema que não possua banco de dados, cada aplicação possui seus próprios arquivos, o que costuma provocar uma redundância considerável nos dados armazenados, seja em papel, ou em arquivos digitais que não tenham o formato e a organização de um banco de dados, o que acaba gerando um desperdício de espaço de armazenamento, pois uma mesma informação poderá ser gravada mais de uma vez em diferentes lugares, entretanto, em um banco de dados, basta gravar a informação uma única vez. O uso de banco de dados tende a eliminar também a inconsistência de dados, pois em vários arquivos, considerando os arquivos de uma empresa por exemplo, pode-se ter diferentes endereços para um mesmo cliente, resultado de mudança de endereço por parte do cliente, atualizado pela empresa em apenas um arquivo. Em um banco de dados, a atualização seria feita uma única vez e estaria disponível para todos os usuários.

3.3 Compartilhamento de dados

A partir de um banco de dados disponível em uma máquina, parcelas isoladas de dados podem ser acessadas por vários usuários simultaneamente, para posteriormente serem utilizadas para diferentes finalidades. Chamamos de acesso concorrente o fato de uma mesma parcela de dados poder ser acessada ao mesmo tempo por vários usuários para as utilizações as quais cada um necessitar.

3.4 Aplicação de restrições de segurança

Um sistema de banco de dados pode restringir o acesso aos dados de acordo com a necessidade de cada usuário. Por exemplo, um funcionário que é encarregado de fazer o controle do estoque de uma fábrica, teoricamente, não precisa ter acesso ao salário de cada funcionário dessa fábrica, já que salário é considerado uma informação com um certo nível de confidencialidade. Portanto, o banco de dados pode restringir os dados que cada usuário vai acessar, tornando mais segura a manipulação dos dados.

4 Estrutura geral de um sistema de banco de dados

Um sistema de banco de dados é composto por um banco de dados e um *software* gerenciador de banco de dados. Esse sistema está dividido em módulos específicos, de modo que todas as necessidades desse sistema possam ser atendidas. Algumas das funções essenciais do sistema de banco de dados podem ser oferecidas pelo sistema operacional e o banco de dados é então construído nessa base. Deve-se portanto, ao construir o banco de dados, considerar a interface entre o sistema de banco de dados e o sistema operacional.

Os componentes funcionais de um sistema de banco de dados podem ser divididos em componentes de processamento de consultas e componentes de administração de memória. Segue abaixo a descrição de cada componente.

4.1 Componentes de processamento de consultas

4.1.1 Compilador DML

Traduz os comandos DML (*Data Manipulation Language*) da linguagem de consulta em instruções de baixo nível, inteligíveis ao componente de execução de consultas. Além disso, o compilador DML tenta transformar a solicitação do usuário em uma solicitação equivalente, mas mais eficiente, buscando assim uma boa estratégia para a execução da consulta.

4.1.2 Pré-compilador para comandos DML

Onde são inseridos em programas de aplicação, que convertem comandos DML em chamadas de procedimentos normais da linguagem hospedeira. O pré-compilador precisa interagir com o compilador DML de modo a gerar o código apropriado.

4.1.3 Interpretador DDL

Interpreta os comandos DDL (*Data Definition Language*) e registra-os em um conjunto de tabelas que contém metadados (dados contendo informações sobre outros dados).

4.1.4 Componentes para o tratamento de consultas

Executam instruções de baixo nível geradas pelo compilador DML.

4.2 Componentes de autorização e integridade

4.2.1 Gerenciamento de autorização e integridade

Testam o cumprimento das regras de integridade e a permissão ao usuário no acesso aos dados.

4.2.2 Gerenciamento de transações

Garante que o banco de dados permanecerá em estado consistente (correto) a despeito de falhas no sistema e que transações concorrentes serão executadas sem conflito em seus procedimentos.

4.2.3 Administração de arquivos

Gerencia a alocação do espaço no armazenamento em disco e as estruturas de dados usadas para representar estas informações armazenadas.

4.2.4 Administração de *buffer*

Responsável pela intermediação de dados no disco para a memória principal e pela decisão de quais dados colocar em memória cache. A vantagem principal na utilização de memória cache consiste em evitar o acesso ao dispositivo de armazenamento - que pode ser demorado - armazenando os dados em meios de acesso mais rápidos.

Além dos componentes de processamento de consulta e de administração de memória, algumas estruturas de dados são exigidas como parte de implementação física do sistema. Essas estruturas estão listadas abaixo:

4.2.5 Arquivo de dados

Armazena o próprio banco de dados.

4.2.6 Dicionário de dados

Armazena os metadados relativos a estrutura do banco de dados. O dicionário de dados é muito usado. Portanto grande ênfase é dada ao desenvolvimento de um bom projeto com uma implementação eficiente no dicionário.

4.2.7 Índices

Proporcionam acesso rápido aos itens de dados que são associados a valores determinados.

4.2.8 Estatísticas de dados

Armazena informações estatísticas relativas aos dados contidos no banco de dados. Essas informações são usadas pelo processador de consultas para seleção de meios eficientes para execução de uma consulta.

Para que tudo que foi dito acima seja possível, ou seja, para criar e gerenciar um banco de dados torna-se necessário um *Software* Gerenciador de Banco de Dados, que será detalhado na próxima seção.

5 *Software* Gerenciador de Banco de Dados

Um banco de dados computadorizado pode ser criado e mantido por um grupo de aplicativos escritos especialmente para essa tarefa. Esse grupo de aplicativos é chamado de *Software* Gerenciador de Banco de Dados.

A expressão *Software* Gerenciador de Banco de Dados é abreviada por SGBD

5.1 O que é um SGBD

Um SGBD é uma coleção de programas que permite ao usuário criar e manter um banco de dados. Ele facilita os processos de definição, onde ocorre a especificação do tipo de dados e das restrições dos dados que serão armazenados, de construção, etapa de armazenamento dos dados em alguma mídia controlada pelo SGBD, de manipulação do banco de dados, onde pode-se incluir várias funções como por exemplo a pesquisa para recuperar um dado específico, a atualização do banco de dados, a geração de relatórios de dados, e outras atividades de acordo com a necessidade do usuário, e finalmente no processo de compartilhamento dos dados entre usuários, permitindo acesso de forma concorrente

5.2 Funcionamento de um SGBD

Um banco de dados requer um grande espaço de armazenamento. Segundo SILBERSCHATZ et al. (1999) os bancos de dados corporativos são medidos em *gigabytes* ou mesmo em *terabytes* de dados. Ele afirma ainda que como a memória central de um computador não pode armazenar essas informações, elas são armazenadas em discos, sendo que os dados são movidos entre os discos e a memória principal de acordo com a necessidade. Porém, a velocidade com que esses dados se movimentam é lenta se for comparada a velocidade da unidade central de processamento. O SGBD estrutura os dados de forma a minimizar o movimento entre os discos e a memória principal e provê a interface entre os dados de baixo nível armazenados num banco de dados e os programas de aplicação e as solicitações submetidas aos mesmos. São tarefas do SGBD.

5.2.1 Interação com o gerenciador de arquivos

Os dados não trabalhados são armazenados no disco usando o sistema gerenciador de arquivos, que geralmente é oferecido por algum sistema operacional. O SGBD traduz os comandos da linguagem de manipulação de dados (DML) em comandos de baixo nível do gerenciador de arquivos. Sendo assim, o SGBD armazena, faz a busca e a atualização dos dados no banco de dados

5.2.2 Cumprimento de integridade

Os dados armazenados num banco de dados precisam satisfazer certos tipos de restrições de consistência. Num campo destinado ao nome dos clientes de uma empresa, pode haver apenas caracteres. Tal restrição será explicitada pelo administrador do banco de dados. O SGBD deve prover funcionalidades para a definição e garantia de restrições.

5.2.3 Cumprimento de segurança

Nem todo usuário do banco de dados precisa ter acesso a todos os dados armazenados no banco de dados. O SGBD pode impor restrições de segurança para acessar os dados.

5.2.4 Cópias de reserva (*backup*) e recuperação

Uma máquina está sujeita a falhas, onde os dados armazenados no banco de dados podem ser perdidos. O SGBD deve detectar as falhas e restabelecer o estado do banco de dados como estava antes da ocorrência das falhas. Isso é feito através de procedimentos de recuperação e cópias de reserva.

5.2.5 Controle de concorrência

O SGBD deve controlar a interação entre usuários simultâneos, pois os mesmos podem atualizar o banco de dados concorrentemente.

Segundo SILBERSCHATZ et al. (1999), muitos SGBDs tem utilitários de bancos de dados. Dentre eles pode-se destacar a transferência de dados de um SGBD para outro através de equipamentos que geram programas apropriados de carregamento e o monitoramento de desempenho que possibilita ao administrador do banco de dados monitorar o desempenho do mesmo e, conseqüentemente, tomar decisões.

5.3 SGBDs para máquinas pessoais

Segundo SILBERSCHATZ et al. (1999), SGBDs projetados para uso em pequenos computadores podem não ter todos os recursos descritos no item 5.2 . Alguns sistemas restringem o acesso a apenas um usuário de cada vez. Outros atribuem ao usuário a tarefa de fazer cópias de reserva ou restrições de segurança. Conseqüentemente, um SGBD como mencionado acima, com menos funções, terá um tamanho menor, o que irá requerer menor quantidade de memória e menor custo.

5.4 As interfaces do SGBD

Podem ser subdivididas em:

5.4.1 Interfaces baseadas em menus para os clientes web ou navegação

Esse tipo de interface apresenta ao usuário listas de opções, os menus, que o guiam durante a formulação de uma pesquisa. Assim, não há necessidade de memorizar comando específicos ou sintaxes de linguagem para realizar uma consulta. Esse tipo de menu geralmente é utilizado nas interfaces para navegação e permitindo que o usuário pesquise o conteúdo de um banco de dados de uma forma exploratória e não estruturada.

5.4.2 Interfaces baseadas em formulários

Exibe um formulário para cada usuário. Não há necessidade de preencher todos os campos do formulário para realizar uma consulta. Normalmente, os formulários são projetados e programados para os usuários iniciantes como interfaces para transações customizadas. Alguns SGBDs contem funcionalidades que permitem que o usuário final construa, interativamente, um formulário na tela.

5.4.3 Interfaces gráficas para o usuário

Uma interface gráfica para o usuário exibe um esquema para o usuário em um formulário diagramático e a consulta pode ser especificada manipulando o diagrama. Em alguns casos, as interfaces gráficas para o usuário utilizam menus e formulários

5.4.4 Interface de linguagem natural

Esse tipo de interface aceita solicitações escritas em inglês ou em outros idiomas e tentam entendê-las. Se a interface conseguir interpretar as solicitações feitas pelo usuário, uma consulta de alto nível é gerada e submetida ao processamento pelo SGDB. Um diálogo é então iniciado com o usuário para esclarecer a solicitação.

5.4.5 Interface para usuários parametrizáveis

Em geral, uma pequena série de comandos adaptados é disponibilizada com o objetivo de minimizar o número de teclas para cada solicitação. As teclas de funções de um terminal, por exemplo, podem ser programadas para iniciar os vários comandos, o que permite ao usuário parametrizável trabalhar com um número mínimo delas

5.4.6 Interfaces para o administrador do banco de dados

Alguns comandos de dados privilegiados são disponibilizados pela maioria dos sistemas de bancos de dados e que apenas administradores de bancos de dados podem utilizá-los. Esses comandos incluem criação de contas, sistema de ajuste de parâmetros, autorizações para criações de contas, mudança de esquemas e reorganização de estruturas de armazenamento do banco de dados.

5.5 Quando não usar o SGBD

Apesar das vantagens no uso do SGBD, ELMASRI e NAVATHE (2005) citam algumas situações em que esse sistema pode envolver custos altos e desnecessários, que normalmente não ocorreriam no processamento tradicional de arquivos. Os altos custos de utilizar SGBD são devidos a:

- Investimentos iniciais altos em *hardware*, *software* e treinamento.
- Generalidade que o SGBD fornece para a definição e processamento de dados.
- Custos elevados para oferecer segurança, controle de concorrência, recuperação e funções de integridade.

ELMASRI e NAVATHE (2005) afirmam ainda que problemas adicionais podem surgir se os projetistas do banco de dados e o administrador do banco de dados não projetarem o banco de dados de maneira adequada ou se a aplicação não for implementada apropriadamente. Sendo assim, indicam o uso de arquivos convencionais nas seguintes circunstâncias:

- O banco de dados e suas aplicações são simples, bem definidas e sem previsão de mudanças.
- Há requisitos de tempo real para alguns programas difíceis de serem atendidos por causa da sobrecarga do SGBD.
- O acesso de múltiplos usuários não é necessário.

5.6 Alguns exemplos de SGBDs

- IBM Informix
- PostgreSQL
- Firebird
- HSQLDB
- IBM DB2
- mSQL
- MySQL
- Oracle
- SQL-Server
- TinySQL
- JADE
- ZODB
- Sybase
- Microsoft Access
- Microsoft Visual Foxpro

6 Modelos de dados

O conceito de modelos de dados é uma coleção de ferramentas conceituais para descrição de dados, relacionamentos de dados, semântica de dados e restrições de consistência. Os vários modelos de dados dividem-se e três diferentes grupos: modelos lógicos baseados em objetos, modelos lógicos baseados em registros e modelos físicos de dados.

6.1 Modelos lógicos baseados em objetos

Os modelos lógicos com base em objetos são usados na descrição de dados no nível lógico e de visões. São caracterizados por dispor de recursos de estruturas bem mais flexíveis e por viabilizar a especificação explícita das restrições de dados. Alguns dos modelos mais conhecidos nessa categoria:

6.1.1 Modelo entidade-relacionamento

Tem por base a percepção do mundo real como um conjunto de objetos básicos, chamados entidades, e o relacionamento entre eles. Uma entidade é um objeto do mundo real, que pode ser identificado por outros objetos. Uma pessoa, por exemplo, pode ser considerada uma entidade, as contas de um banco, também são entidades e entre essas duas entidades, há um relacionamento. O relacionamento é uma associação entre as entidades.

6.1.2 Modelo orientado a objetos

Tem por base um conjunto de objetos. Um objeto contém valores armazenados em várias instâncias dentro do objeto. Instância é o conjunto de informações contidas em um determinado banco de dados em um dado momento. Um objeto também tem um conjunto de códigos, chamados métodos, que operam esse objeto. Os objetos que contém os mesmos tipos de valores e os mesmos métodos são agrupados em classes. Uma classe pode ser vista como uma definição de tipo para objetos.

Essa combinação compacta de dados e métodos abrangendo uma definição de tipos é similar ao tipo abstrato em uma linguagem de programa. O único modo pelo qual um objeto pode conseguir acesso aos dados de outro objeto é por meio do método desse outro objeto.

6.2 Modelos lógicos baseados em registros

São usados para descrever os dados no nível lógico e de visão. São usados tanto para especificar a estrutura lógica do banco de dados quanto para implementar uma descrição de alto nível.

6.2.1 Modelo relacional

Nesse modelo o banco de dados é representado como uma coleção de relações. Pode-se pensar uma tabela como uma coleção de relações, onde cada linha representa um conjunto de dados relacionados entre si. Os dados de uma linha representam fatos do mundo real

6.2.2 Modelo hierárquico

A organização dos dados no modelo hierárquico é do tipo árvore. A estrutura de árvore pode ser pensada como uma raiz que se distribui em ramos e folhas. Uma raiz tem vários ramos e

e cada galho tem várias folhas. O caminho raiz-galho-folha é único, não existindo duas formas diferentes de chegar a uma folha a partir da raiz.

Tal organização lembra, como o próprio nome do modelo sugere, que existe uma determinada hierarquia entre os dados.

6.2.3 Modelo de rede

Esse modelo é uma extensão do modelo hierárquico, porém difere do hierárquico no fato de que no último, um registro filho pode ter apenas um registro pai, já no primeiro, pode ter vários registros pai.

6.3 Modelos físicos de dados

Os modelos físicos de dados são usados para descrever os dados no nível mais baixo. Esses modelos captam os aspectos de implementação do sistema de banco de dado.

7 Tipos de Banco de Dados

7.1 Banco de Dados Relacional

O banco de dados relacional segue o modelo Relacional (o modelo relacional está descrito na seção modelos de dados). Este tipo de banco de dados surgiu por volta de 1970.

A linguagem padrão dos Bancos de Dados Relacionais é a *Structured Query Language*, ou simplesmente SQL como é mais conhecida. O termo relacional é aplicado aos próprios dados e ao SGDB. Foi desenvolvido para prover acesso facilitado aos dados, possibilitando que os usuários utilizassem uma grande variedade de abordagens no tratamento das informações. Pois, enquanto em um banco de dados hierárquico os usuários precisam definir as questões de negócios de maneira específica, iniciando pela raiz do mesmo, no banco de dados relacional os usuários podem fazer perguntas relacionadas aos negócios através de vários pontos. A principal proposição do modelo relacional é que todos os dados são representados como relações matemáticas.

Um banco de dados relacional consiste em uma coleção de tabelas, cada uma das quais com um nome único. Os dados de um banco de dados relacional (BDR) são armazenados em tabelas, que por sua vez, nada mais são do que estruturas de linhas e colunas. Uma linha em uma tabela representa um relacionamento entre um conjunto de valores. Um banco de dados podem possuir várias tabelas. Segundo SILBERSCHATZ et al. (1999) uma vez que uma tabela é uma coleção de relacionamentos, há uma estreita relação entre o conceito de tabela e o conceito matemático de relação, a partir dos quais se origina o nome desse modelo de dados.

7.1.1 Exemplos de banco de dados, descrição de algumas tabela e seus possíveis atributos

- Cliente (Cliente, Taxa, Nome, Endereço, Cidade, Estado, CEP, Telefone)
- Pedido de compra (Número do pedido, Cliente, Fatura, Data do pedido, Data prometida, *Status*)
- Item do pedido (Número do pedido, Número do item, Código do produto, Quantidade)
- Nota fiscal (Número da nota, Cliente, Número do pedido, Data, *Status*)
- Item da nota fiscal (Número da nota, Número do item, Código do produto, Quantidade vendida)

Segundo ainda SILBERSCHATZ et al. (1999), esse tipo de banco de dados é útil para lidar com os problemas de bancos de dados para aplicações administrativas e comerciais sendo a tecnologia mais difundida na área.

7.2 Banco de dados Hierárquico

Trabalha com os dados e relacionamentos como uma coleção de registros relacionados por ligações. As ligações entre os registros podem ser chamadas *links*. Nesse tipo de banco de dados os dados estão agrupados sob a forma de árvores. Cada registro é uma coleção de atributos (campos), cada um dos quais contendo somente uma informação; uma ligação é a associação entre dois registros. Por exemplo: em uma dada base de dados comercial, uma encomenda (registro) é de um único cliente.

Os registros são organizados sob a estrutura de árvore com raiz. A raiz da árvore é a origem dos dados daquela árvore. A árvore é dividida em nós, onde cada nó é um registro. Entre dois nós, ou seja dois registros, não podem existir outros registros. A relação entre esses nós é semelhante a estrutura de árvore, ao sair da raiz e chegar a uma folha, existe apenas um caminho para chegar ao destino final. Uma base de dados hierárquica pode ter várias árvores.

Um exemplo simples desse tipo de banco de dados seria a relação pai-filho, onde o pai é a raiz, e o filho, o nó. Dessa relação, podemos perceber que a raiz pode ter muitas folhas, ou seja, o pai pode ter muitos filhos, mas um filho não pode vários pais. Quando essa relação hierárquica é quebrada passa-se a ter um banco de dados de rede.

7.3 Banco de dados de rede

Enquanto no modelo relacional os dados e os relacionamentos entre dados são representados por uma coleção de tabelas, o modelo de rede representa os dados por coleções de registros e os relacionamentos entre dados são representados por *links*. O banco de dados de rede é uma coleção de registros que são conectados uns aos outros por meio de *links*. Cada registro é uma coleção de campos. Um *link* é uma associação entre exatamente dois registros.

Sua organização se assemelha a dos bancos de dados hierárquico, porém difere no sentido de que enquanto no hierárquico, cada registro filho pode ter apenas um registro pai, no modelo de rede, um registro filho pode ter vários registros pai, permitindo criar conexões mais complexas.

Por ser um modelo de estrutura mais completa, possui as propriedades básicas de registros, conjuntos e ocorrências, utilizando linguagem de definição de banco de dados e linguagem de manipulação de dados, o que permite uma evolução mais eficiente do modelo. Para exemplificar pode-se considerar um banco de dados representando o relacionamento conta-cliente de um sistema bancário. A associação entre conta e cliente é o que chamamos de *link*.

7.4 Banco de dados Objeto-Relacional

Os bancos de dados tradicionais como os bancos relacionais, de rede e hierárquicos foram bem sucedidos na maioria das aplicações comerciais de banco de dados. Porém, bancos de dados para projetos de engenharia e manufatura, experimentos científicos, telecomunicações, e outros, podem ser mais complexos. Os bancos de dados orientados a objeto foram propostos para atender a essas necessidades de aplicações mais complexas.

Algumas diferenças desses bancos em relação aos tradicionais são as estruturas complexas para objetos, transações de longa duração, novos de tipos de dados para armazenamento de imagens ou textos longos. Esse tipo de banco permite ao projetista especificar tanto a estrutura dos objetos complexos quanto as operações que podem ser feitas sobre eles.

Outro motivo desse tipo de banco ter sido criado é o fato de linguagens de programação orientadas a objeto serem cada vez mais utilizadas no desenvolvimento de aplicações de *software*.

8 Linguagens de banco de dados

Podemos dividir em duas categorias os tipos de linguagens oferecidas por um sistema de banco de dados. Uma delas para expressar consultas e atualizações e outra específica para os esquemas do banco de dados.

8.1 Linguagem de definições de dados

Do inglês, *Data Definition Language* (DDL), é um tipo de linguagem especial onde um esquema de dados é especificado por um conjunto de definições. É uma linguagem formal usada para descrição de estruturas de dados.

Os parâmetros de linguagem, após serem compilados, são armazenados em um dicionário de dados. O dicionário de dados é um arquivo de metadados (dados a respeito de outros dados), consultado sempre antes que um dado seja modificado.

8.2 Linguagem de manipulação de dados

Do inglês, *Data Manipulation Language*. A manipulação de dados envolve a recuperação de informações armazenadas no banco de dados, a inserção de novas informações, a remoção e a modificação de informações do banco de dados, tendo como objetivo proporcionar uma interação eficiente entre usuário e máquina. A linguagem de manipulação de banco de dados viabiliza o acesso ou a manipulação dos dados de forma compatível ao modelo de dados apropriados. Definem-se algoritmos que permitem o acesso eficiente aos dados.

Existem, basicamente, dois tipos de DMLs, as procedurais, onde o usuário deve especificar os dados que quer obter e como obtê-los, e as não procedurais, onde o usuário precisa apenas especificar quais os dados quer obter.

9 Usuários do banco de dados

Segundo SILBERSCHATZ et al. (1999), a meta básica de sistema de banco de dados é proporcionar um ambiente de recuperação de informações e para o armazenamento de novas informações no banco de dados. Ele classifica os usuários em 4 tipos, diferenciando-os por suas expectativas.

9.1 Usuários finais causais

Acessam o banco de dados ocasionalmente, buscando informações diferentes a cada acesso. Esse tipo de usuário costuma usar uma linguagem sofisticada de consulta ao banco de dados para especificar suas solicitações. Normalmente são gerentes de nível médio ou elevado ou outros profissionais com necessidades ocasionais.

9.2 Iniciantes ou usuários finais parametrizáveis

A maioria dos usuários finais de bancos de dados. Acessam constantemente o banco de dados fazendo consultas ou atualizações usando tipos de consulta e atualização padronizada, que tenham sido cuidadosamente programadas e testadas. Podem ser os caixas de bancos, os funcionários responsáveis pela reserva de vôos, hotéis e locação de carros, funcionários de correios atualizando a situação de pacotes recebidos ou em trânsito e outros.

9.3 Usuários mais sofisticados

Nessa categoria incluem-se engenheiros, cientistas, analistas de negócios e outros profissionais que se familiarizam com as facilidades do SGBD e as usam para implementar ações que atendam as suas solicitações complexas.

9.4 Usuários autônomos

Usuários que mantêm um banco de dados pessoal por meio do uso de pacotes de programas prontos. Esses programas prontos possuem interfaces gráficas ou programas baseados em menus normalmente fáceis de usar. Um exemplo disso é o usuário de um pacote para cálculo de imposto de renda que armazena seus dados financeiros pessoais para o pagamento do imposto de renda.

10 Administrador do banco de dados

Do inglês, *Data Base Administrator (DBA)*. Para bancos de dados de tamanho considerável, onde várias pessoas irão acessá-lo e usar os recursos por ele disponibilizados, torna-se necessário um administrador para organizar e gerir os recursos oferecidos pelo banco de dados. No ambiente de banco de dados, os principais recursos são o banco de dados, o SGBD e os *softwares* relacionados. É de responsabilidade do DBA a autorização para o acesso ao banco, a coordenação e monitoração do seu uso e por adquirir recursos de *software* e *hardware* quando se tornar necessário. O DBA é responsável ainda por problemas com brechas de segurança ou quando o tempo de resposta do sistema é muito grande. Em grandes organizações o DBA possui ainda suporte de assistentes.